

Università degli Studi di Trento Corso di Programmazione I – E3: 145430/145935 3° Appello – 07/06/2018 – a.a. 2017/2018		PC N°						
COGNOME NOME MATRICOLA <table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> </tr> </table>								

Completare le voci **Cognome**, **Nome** e **Matricola**, se tali voci saranno mancanti o non-leggibili elaborato **NON** verrà corretto.

NON è consentito utilizzo di alcun strumento cartaceo, elettronico, informatico etc. durante la prova.

Fa parte della valutazione la leggibilità del codice C/C++.

Scrivere le risposte su questo foglio e riconsegnare. I risultati verranno pubblicati su DOL
Tempo a disposizione: 120 minuti.

PARTE TEORICA: 12 PUNTI – PUNTEGGIO MINIMO 6 PUNTI

[Domanda 1]

[3.1 - punti 1]

Data la seguente sequenza di valori

4, 4, 2, 6, 3, 5, 5

costruire un albero binario di ricerca (BST), mettendo i valori uguali nel sottoramo di SX.

--

[3.2 - punti 3] Per l'albero così realizzato inoltre calcolare

Cammino							
Altezza							
Visita pre-ordine							
Visita post-ordine							
Visita in-ordine							

[Domanda 2 - punti 4] data una sequenza SQ di numeri naturali memorizzata in un array creare una funzione “CercaSQPrimo” a cui passare l'array SQ

bool CercaSQPrimo (int sq[], int len)

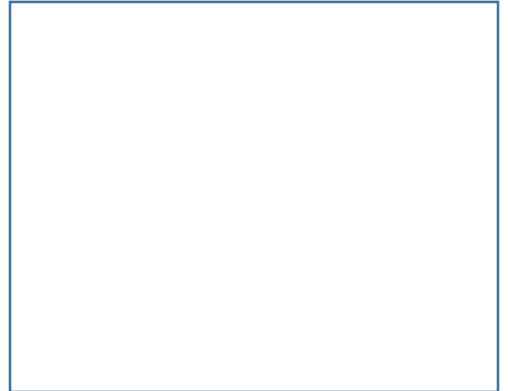
che determina se esiste una sotto sequenza di numeri inseriti nell'array tale per cui la sua somma è un numero primo. In questo caso ritornare true altrimenti false (si supponga esista una funzione bool n_primo(int n) che restituisce true se in numero è primo, false altrimenti).

[Domanda 3 - punti 4] Dato il seguente codice:

```
#include <stdio.h>
#include <stdlib.h>
#define N 6
int main () {
    int i, j, temp;
    int seq[n];
    i = 0;
    while (i < N) {
        // inserimento dati matricola
        scanf("%d",&seq[i++]);
    }
    j = 1;
    while (j < N){
        temp = seq[j];
        i = j -1;
        while (i >= 0 && seq[i] > temp){
            printf("temp=%d\n",temp);
            seq[i+1] = seq[i];
            i = i -1;
        }
        seq[i+1] = temp;
        j += 1;
    }
    printf("il risultato dell'algorithm e': ");
    for( i = 0; i < N; ++i){
        printf("%d",seq[i]);
    }
}
```

[3.1] Considerando l'esecuzione del programma su calcolatore, si scriva esattamente l'output mostrato a video.

[3.2] Si descriva brevemente il compito eseguito dall'algorithm.



[3.3] Si calcoli la funzione $T(n)$ nel caso peggiore e la relativa notazione Teta ($\Theta(n)$).

PARTE PRATICA: 21 PUNTI – PUNTEGGIO MINIMO 11 PUNTI

[A - punti 3] Inserire i seguenti tipi in un file **dati.h** e implementare i metodi di stampa ed i costruttori:

```
typedef enum Tpacco { ORDINARIO, CELERE, PRIORITY } Tpacco;  
  
typedef struct Tspedizione{  
    char destinatario[20];  
    bool assicurato;  
    float peso;  
    Tpacco tipoPacco;  
    ..Tspedizione () { /* da implementare */ }  
    //implementare il metodo di stampa → formato: vedi punto E  
} Tspedizione;  
  
typedef struct TcodaFIFO {  
    Tspedizione dato;  
    TcodaFIFO* next;  
    TcodaFIFO () { /* da implementare */ }  
    TcodaFIFO (Tspedizione d) { /* da implementare */ }  
    TcodaFIFO (TcodaFIFO* n, Tspedizione d) { /* da implementare */ }  
    //implementare il metodo di stampa  
} TcodaFIFO;
```

[B - punti 1] Modificare il file **main.cpp** includendo il file **dati.h** ed inserendo il seguente codice nel **main**:

```
TcodaFIFO* p[3] = {NULL, NULL, NULL};  
Tspedizione s;  
for(int i=0; i<5; i++) { creaSpedizione(&s); addPacco(p,s); }  
stampaSpedizioni(p);  
delPacchi(p, ORDINARIO);
```

[C - punti 4] Implementare una funzione **creaSpedizione** che riceve in input per referimento una variabile di tipo **Tspedizione s** e la inizializza con:

- campo **tipoPacco** valore casuale tra ORDINARIO, CELERE, PRIORITY;
- campo **peso** valore letto da tastiera, tra 200.00 e 5000.00, con controllo input.
- campo **assicurato** pari a true se tipoPacco è CELERE o PRIORITY, false altrimenti;
- campo **destinatario** letta da tastiera.

[D - punti 4] Implementare una funzione **addPacco** che riceve come input una variabile di tipo array **TcodaFIFO* p** e variabile di tipo **Tspedizione s** e la inserisce nella lista **p[x]** (con **x** valore casuale tra 0 e 2) (modalità *insertFirst*).

[E - punti 4] Implementare una funzione **stampaSpedizioni** che riceve in input una variabile di tipo array **TcodaFIFO* p** e stampa il contenuto delle liste utilizzando opportunamente il metodo **stampa di TcodaFIFO**.

Nella forma: **destinatario PESO=peso PACCO=tipoPacco**

[F - punti 3] Implementare una funzione **delPacchi** che riceve in input una variabile di tipo array **TcodaFIFO* p** ed un tipo di pacco **tp**: estrae (modalità *getFirst + removeFirst*) tutti gli elementi dalla coda **p[0]** e salva su file i soli pacchi di tipo **tp**.

Inserire i dati in un file denominato “**pacchiOrdinari.txt**”, nella forma

destinatario PESO=peso

È possibile creare strutture, funzioni, metodi aggiuntivi per realizzare gli esercizi proposti.

Il codice commentato NON verrà corretto!